“Give me what I want” - Enabling Complex Queries on Rich Multi-Attribute Data

Josie McCulloch (Student Member), Christian Wagner (Senior Member), Khaled Bachour, Tom Rodden
School of Computer Science, University of Nottingham, Nottingham, United Kingdom
psxjm5; christian.wagner; khaled.bachour; tom.rodden@nottingham.ac.uk

***Abstract*—Consumer and more generally, human preferences are highly complex, depending on a multitude of factors, most of which are not crisp, but uncertain/fuzzy in nature. Thus, user selection amongst a set of items is dependent on the complex comparison of items based on a large number of imprecise item-attributes such as price, size, colour, etc. This paper proposes the mechanisms to underpin the digital replication of such complex preference-based item selection with the view to enabling improved digital item search and recommendation systems. For example, a user may query “I would like a product of similar size but at a cheaper price.” The proposed method involves splitting query-attributes into two categories; those to remain similar (e.g., size) and those to be changed in a specific direction (e.g., price - to be lower). A combination of similarity and distance measures is then used to compare and rank recommendations. Initial results are presented indicating that the proposed method is effective at ranking items according to intuition and expected user preferences.**

I. Introduction

Consumer, and more generally, human preferences are highly complex, depending on a multitude of factors, most of which are not crisp, but uncertain/fuzzy in nature. Thus, everyday choices, such as “which type of cake should I have?”, can be thought of as complex, context dependent queries where individual items (e.g., cakes) are compared based on a number of attributes such as *sweetness, size*, etc., resulting in a rank ordering of available items and, finally, the selection of a specific item.

This paper presents a method of handling such complex queries on uncertain/fuzzy data with a view to enable comprehensive preference-based ordering of a collection of potential choices. In practice, when a user creates a query to find a desired item, it is common for the user to only have a vague idea of his or her preferences. For example, a user may have an idea of an approximate price they may wish to pay and have specific aspects they are looking for, e.g., “I would like a cake of a similar price but more fruity than chocolate cake.” Attributes such as *price* and *fruity* are often best captured by fuzzy sets (FS) as they are subjective. Fig. 1 offers an example of FSs describing how much two different cakes are considered fruity. Using the IAA approach [1], these FSs have been constructed from real data which are both non-normal and non-convex as a result of disagreement between different people. In searching for a cake more fruity than chocolate cake, the lemon cake is a suitable recommendation because its FS is rated higher than chocolate.

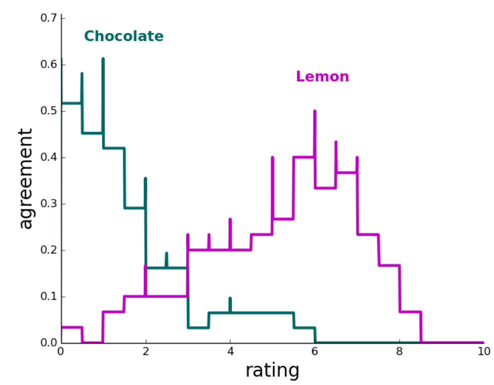


Fig. 1. Fuzzy sets representing how much two different cakes are fruity.

Given that the data (price-range, fruitiness) are imprecise, it follows that suitable queries should be based on the in-depth joint comparison of all relevant attributes of all available choices. Such queries then enable a user to search for items using a Computing with Words [2] approach and enable human-like and personalised, but automatic, recommendation systems.

Generally, comparisons of multi-attribute data for the purpose of classification, recommendation and ranking have become a popularly researched topic in recent years as advances in online purchasing have increased the desire for personalised recommendations. A common application is content-based recommendations, which involves the recommendation of items based on the preferences of a user [3], [4], [5], [6]. A user profile is held, which often contains historical data such as previous purchases, and is used to calculate the user's preferences. Items can then be matched against the user's preferences, and the highest matching items are recommended.

For example, Hsu [7] uses clustering and association rules to recommend suitable lessons for students studying the English language. Recommendations are calculated by considering the past study behaviours of the individual students. Wang et al. [8] also used clustering and association rules to recommend products to customers based on their previous purchases.

Though content-based recommendations are common, one issue with using user profiles is that the approach is not ideal when there is little information regarding a user's precise pref-

This work was partially funded by the EPSRC's Towards Data-Driven Environmental Policy Design grant, EP/K012479/1 and the RCUK's Horizon Digital Economy Research Hub grant, EP/G065802/1.

erences. With this problem in mind, Cao and Li [9] developed a data-mining and fuzzy logic based recommendation system for selecting items which are not frequently purchased and, as a result, have no sales history from which recommendations can be based.

This paper, likewise, focuses on providing recommendations where there is little to no absolute data available regarding the user's preferences. The initial information, on which recommendations will be based, may be from the first item(s) which the user views or selects. It is then based on the user's relative preferences in relation to this/these item(s) that items are ranked. Thus, as in the above example, a user may wish to keep some attributes of the initial item the same (e.g., price) and change others (e.g., fruitiness). Using this, new items are compared against the initial item. Ranking is used to indicate a high-level order in preference, while continuous relative similarity is maintained as a richer (rather than an ordinal notation) model of the preference order.

A measure of similarity is used to compare the attributes which the user would like to keep the same, and a measure of distance is used to compare attributes which are to be changed (either increased or decreased). A directional distance measure is used to ensure that the correct direction is taken into account when measuring distance. For example, if a user wants to find a cake which is more fruity, then the distance measure must not only rank alternatives in order of how much change there is in the attribute fruity, but must also ensure that the direction of fruity from the initial comparison to the recommended item is a positive one, ensuring that the attribute is increased.

The above two comparisons, by similarity and distance, are fused together into a single value by using an ordered weighted average operator (OWA) [10]. By fusing the similarity and distance between items, one is able to understand the comparison through a single value, rather than needing to interpret the similarity and distance of different attributes separately, which is not trivial in all cases.

The next section provides a background on the measures used to compare FSs throughout this paper, followed by a demonstration to show the advantages of using both similarity and distance in making recommendations in Section III. Following this, the proposed method of recommending items based on multi-attribute data is presented in Section IV, after which demonstrations are given in Section V. Finally, conclusions are presented in Section VI.

II. Background

The following introduces FSs and the measures used to compare FSs within the proposed recommendation process. These measures include similarity, directional distance, an OWA operator, and a "proximity" measure based on the former two measures using an OWA.

A. Fuzzy Sets

A FS F within the universe of discourse X may be represented as a set of ordered pairs of an element x and its membership value within F, denoted $\mu_F(x)$, i.e.

$$F = \{(x,\ \mu_F(x)) \mid x \in X\}$$

A FS may also be defined as a collection of its alpha-cuts (α-cuts) [11]. An α-cut of a FS F is a crisp set defined as

$$F_\alpha = \{x \mid \mu_F(x) \geq \alpha, \alpha \in [0,1]\} \tag{1}$$

The FS F can be represented by its alpha cuts as [11]

$$F = \int_0^1 \alpha F_\alpha \tag{2}$$

where $\int_0^1$ is the union of all F_α ranging from 0 to 1.

B. Comparing Fuzzy Sets

Similarity and distance are important measures in comparing two FSs. Measuring similarity involves taking vertical slices and comparing the membership values of the elements within the FSs. Measuring distance, however, generally involves taking α-cuts and comparing the elements contained within the FSs. The distance between α-cuts may be weighted by the membership values of the elements belonging to the α-cuts.

1) Similarity: A similarity measure $s : F(X) \times F(X) \to [0,1]$ calculates how close the membership values are of two FSs for each element. A common similarity measure on FSs is the Jaccard measure which is calculated as

$$s(A,B) = \frac{\sum_{i=1}^{n} min(\mu_A(x_i),\ \mu_B(x_i))}{\sum_{i=1}^{n} max(\mu_A(x_i),\ \mu_B(x_i))} \tag{3}$$

where A and B are FSs and n is the total number of discretisations along the x-axis. The result 0 indicates that A and B are disjoint and the value 1 indicates they are identical.

A measure of dissimilarity may be obtained by using the complement of similarity [12]; that is,

$$s'(A,B) = 1 - s(A,B) \tag{4}$$

2) Distance: A distance measure $d : F(X) \times F(X) \to \mathbb{R}^+$ calculates the difference between the elements contained within two FSs, where the result 0 indicates identical fuzzy sets. This paper uses Chaudhur and Rosenfeld's [13] distance measure, which is as follows

$$d(A,B) = \frac{\sum_{i=1}^{m} y_{\alpha_i}\ h(A_{\alpha_i}, B_{\alpha_i})}{\sum_{i=1}^{m} y_{\alpha_i}} \tag{5}$$

where the y-axis is discretised into m points $(y_1, y_2, ..., y_m)$, A_{α_i} is the non-fuzzy α-cut of the FS A at y-coordinate y_{α_i} and h is commonly represented by the Hausdorff measure. However, instead of the Hausdorff measure, this paper uses a directional measure. The proposed directional measure results in a positive value for $h(A,B)$ where $A < B$ and a negative value for $h(A,B)$ where $A > B$. Essentially, the sign of the function represents the direction taken along the x-axis when travelling from A to B.

Generally, for a normal, convex FS, the α-cut A_{α_i} may be represented as a continuous interval obtained from the left-most and right-most values within A_{α_i}. Given this, the proposed directional distance between two α-cuts is given as:

$$h(\bar{A}, \bar{B}) = \frac{1}{2}(\bar{B}_l + \bar{B}_r - \bar{A}_l - \bar{A}_r) \tag{6}$$

where $\bar{A}$ is a continuous interval from A_{α_i} with left and right boundaries $\bar{A}_l$ and $\bar{A}_r$, respectively, and is denoted $[\bar{A}_l, \bar{A}_r]$. Likewise, $\bar{B}$ is the continuous interval $[\bar{B}_l, \bar{B}_r]$. Using (6) gives the average distance between the left and right boundaries of the intervals. This is akin to the directional distance proposed by Yau and Wu [14], however the proposed approach weights each distance h by the value of α. Note that when using (6) the distance measure returns a value within $\mathbb{R}$, whereas a non-directional distance measure would return a value in $\mathbb{R}^+$.

The result of (5) may be normalised as [10]

$$d_n(A, B) = \frac{d(A, B)}{\lambda} \tag{7}$$

where λ is the largest possible distance within the universe of discourse. For a finite universe of discourse X, described as $\{x_1, x_2, ..., x_n\}$, λ will be $x_n - x_1$.

To provide an efficient and thorough comparison of two FSs, the resulting values of a similarity measure and distance measure may be combined together into a single value [10]. This is useful because both similarity and distance are at times not useful individually; see also Section III. By doing this, one can use a single value to determine both how much two FSs overlap and how much distance there is between their membership values. To achieve this value, the similarity and distance of FSs is fused using an OWA operator. The next two sections introduce the OWA and a fusion of similarity and distance using an OWA.

3) Ordered Weighted Average Operator: An OWA takes a list of values which are sorted into descending order, and an ordered set of weights $w = \{w_1, w_2, ..., w_n\}$, for which $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$. The input values are sorted into descending order and each value is multiplied by the corresponding weight. Thus, for a given list of objects $a_1, a_2, ..., a_n$ and weights $w_1, w_2, ..., w_n$, the OWA is calculated as [15]:

$$f(a_1, a_2, ...a_n) = w_1 b_1 + w_2 b_2 + ... + w_n b_n \tag{8}$$

where b_i is the i_{th} largest element in the collection $a_1, ..., a_n$.

4) Combined Similarity and Distance: Similarity and distance measures can be fused together using an OWA (8) to create an overall comparison of the proximity of two FSs [10]. This combined proximity measure, $c : F(X) \times F(X) \rightarrow [-1, 1]$, may be achieved by averaging dissimilarity and normalised distance as follows [10]:

$$c(A, B) = \begin{cases} f\big(s'(A, B),\ d_n(A, B)\big), & d_n(A, B) \geq 0 \\ f\big(-s'(A, B),\ d_n(A, B)\big), & \text{otherwise} \end{cases} \tag{9}$$

where F is an OWA (8), s' is a dissimilarity measure (4) and d_n is a normalised directional distance (7).

We use the weights $w = \{0.7, 0.3\}$ as established in [10], giving the largest value of $s'(A, B)$ and $d_n(A, B)$ the weight 0.7, and the smallest value is assigned the weight 0.3. Note that the absolute values of the measures are used when assigning the weights, thus a measure of -0.45 is considered larger than a measure of 0.3.

Using (9), the result $c(A, B) = 0$ indicates that A and B are identical FSs, and the result 1 or -1 indicates two crisp sets which have the maximum distance possible in the given universe of discourse. As with the directional distance between intervals (6), $c(A, B) > 0$ indicates that $B > A$ and $c(A, B) < 0$ indicates that $B < A$.

Having introduced the measures which will be used to compare FSs, the next section demonstrates the benefits of using combined similarity and distance to find recommendations.

III. Similarity and Distance in Recommendations

While we have given examples of a user searching for cakes in the introduction, another example of a complex, imprecise query is: "find a restaurant similar to this one I like but with a higher quality of food." In this query, the user has stated that when comparing restaurants to "this one I like", all attributes except *quality of food* should be as similar as possible, whilst the attribute *quality of food* must not only be different but, specifically, it should be rated higher (i.e. the FS should contain higher valued elements). In other words, similarly good but slightly worse (or equally good) items should not be returned.

To achieve this, a similarity measure can be used to compare each attribute of each item, except for the attribute *quality of food*. To find a restaurant with a higher quality of food, a directional distance measure can be used to find the restaurant with the greatest distance in food quality, where the direction of that distance is a positive one. The following shows why it is important to use both similarity and distance to solve this query by demonstrating that using only one measure does not always provide appropriate results for a given query. These demonstrations also show the advantages of using the combined "proximity" measure (9) to find recommendations.

To find items which are similar to each other, a similarity measure is often an ideal approach. Although a distance measure can provide details on the proximity of two FSs, it is not ideal for determining how well two FSs share the same elements with similar membership values. For example, consider the three FSs A, B and C in Fig. 2(a). The distance between pairs (A, B) and (A, C), as shown in Table I, are identical, however, the similarity is different. Thus, demonstrating that using a distance measure to find the shortest distance between pairs of FSs is not always an ideal substitute for a similarity measure. However, using the results of the combined measure (9) in Table I, it is possible to see which pair is the most similar by the smallest result from (9).

Likewise, distance is important for comparing the attributes of items to see if an attribute is better or worse for a given item. Similarity is not an ideal substitute as, although it can be used to determine that FSs are dissimilar and therefore somewhat distant, it is not an ideal approach for determining the degree of distance between FSs in terms of the elements

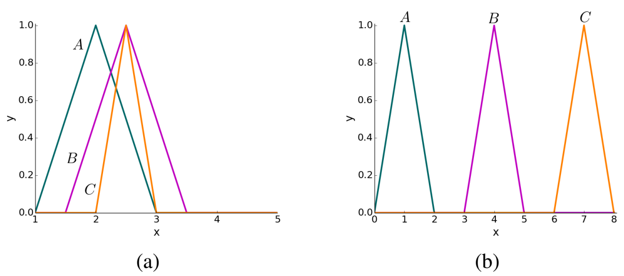


Fig. 2. Three FSs A, B and C.

TABLE I
COMPARISON OF SIMILARITY AND DISTANCE AGAINST THE FSS IN FIGURE 2(A).

Method	(A,B)	(A,C)
similarity (3)	0.394	0.276
distance (5)	0.5	0.5
combined (9)	0.462	0.544

TABLE II
COMPARISON OF SIMILARITY AND DISTANCE AGAINST THE FSS IN FIGURE 2(B).

Method	(A,B)	(A,C)
similarity (3)	0.0	0.0
distance (5)	3.0	6.0
combined (9)	0.79	0.88

contained within these sets. For example, consider Fig. 2(b) which contains three FSs A, B and C, the comparisons of which are shown in Table II. Pairs (A, B) and (A, C) are both disjoint and thus have a similarity of 0, but this does not show which pair has the greatest distance. Using the distance measure, however, it is clear that A and C are the furthest pair. Note, this is also evident when using the combined measure.

It is therefore not possible to determine how far apart FSs are by measuring their similarity, making it an unsuitable substitute for distance in this application. Additionally, it is necessary to use distance to find the direction between FSs, even if only by comparing the centroids, as a similarity measure cannot be used to determine if one FS is to the left or right of another.

One final example shows the results obtained when ranking with the combined measure. Consider the FSs A and B in Fig. 3(a) (and in Fig. 3(b)), both of which share the same mean and are symmetrical but have different standard deviations. To the left of the mean, B can be said to be to the left of A because it contains elements that are lower than those in A. However, from the right of the mean, B can be said to be to the right of A because it contains values that are higher than those in A. Therefore, B can be described as being both to the left and right of A. In this special case, the distance between A and B is 0. This is only true when the distance both sides of the mean are equal. However, as (9) also takes into account the dissimilarity of A and B, the results of $c(A, B)$ for Fig. 3(a) and Fig. 3(b) results in different values (0.337 and 0.454, respectively).

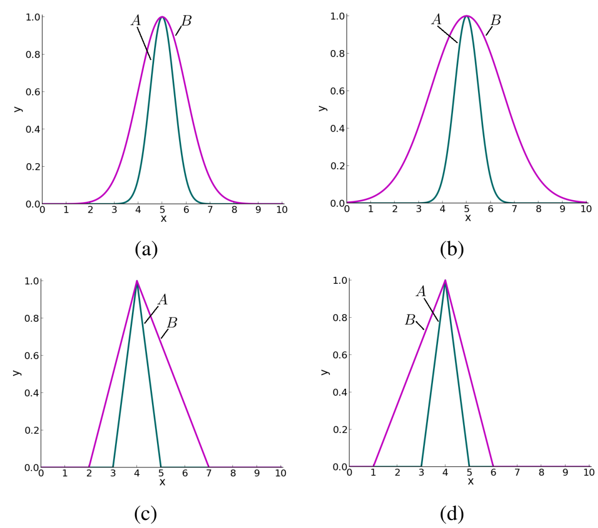


Fig. 3. Pairs of heavily overlapping FSs.

In contrast, the FS B in Fig. 3(c) is described as being to the right of A as the biggest distance occurs to the right of the mean rather than the left. In this case, $c(A, B) = 0.409$. However, in Fig. 3(d), B is said to be to the left of A because this is where the most significant distance is located. In this case, $c(A, B) = -0.409$.

Having demonstrated the benefits of using combined similarity and distance to calculate recommendations, the next section goes through the proposed recommendation process.

IV. PREFERENCES IN MULTI-ATTRIBUTE DATA

Consider a group of items (e.g., restaurants) which will be compared, ranked and recommended. Each restaurant is described by its attributes (e.g., *quality of food*, *quality of service*, etc.), and each attribute is represented by a FS. Fuzzy sets are useful here because they capture the uncertainty in user opinions [1]. In the following, the set of all attributes for the item i is given as $F_{iQ} = \{F_{i1}, f_{i2}, ..., F_{in}\}$, where f_{iq} is the FS that represents the attribute q of item i, and Q represents the set of all attributes $\{q_1, ..., q_n\}$.

Using these FSs, comparisons can be made through (9). This compares two FSs by combining information about the dissimilarity in their membership values, and the distance between the elements belonging to the FSs. Using this approach, it is straightforward to find items which are similarly rated by finding attributes (FSs) which are similar. As stated earlier, $c(A, B) = 0$ if $A = B$, therefore to find similar items, e.g. restaurants, one can compare each attribute for each restaurant and choose the restaurant whose average comparison among all attributes is the closest to 0.

However, in a recommendation system, one may wish to specify a change in one or more attributes. For example, one may ask for "a similar restaurant to this but with a nicer ambience." In this case, it is still desired that all attributes other than *ambience* have an average as close to 0 as possible. However, it is desired that the comparison between the rating *ambience* of two restaurants results in a value as close to 1 as possible. If, in contrast, the query is "a similar restaurant to this but less expensive" then it is desired that the comparison on *expensive* results in a value as close to -1 as possible.

A. Proposed Method of Recommendation

The following outlines the steps taken to compare a group of items, where the query may be in the form of the above examples. Each item is represented by the collection of its fuzzy attributes; i.e. $F_{iQ} = \{F_{iq} \mid \forall q \in Q\}$. As a variety of notations are used to describe the proposed process, Table III provides a reference of each notation.

1) For each attribute, compare between items using (9). First, compare all attributes except the attribute you wish to change as follows

$$\delta(j,i,p) = \frac{1}{t_Q - 1} \sum_{q \in Q; q \neq p} |c(F_{jq}, F_{iq})| \tag{10}$$

where j is the initial item to be compared against, i is the item being compared with j, p is the attribute that is to be changed, and t_Q is the total number of attributes. Note that (10) is calculated once for each item F_i where $i \neq j$.
The resulting value represents the average comparison among all attributes, describing the degree to which the item i is not a suitable comparison against the item j. This result is within $[0, 1]$ and will be referred to as the degree of *unfitness*. Note that, at this stage, the direction of the results is unimportant, therefore the absolute value from the combined proximity measure is used.

2) Next, compare the item of interest p, taking into account the desired direction. This is shown in (11) where ϵ indicates the desired direction from c. ϵ is 1 if one wishes $c > 0$, or ϵ is -1 if one wishes $c < 0$. Thus, at this stage, the direction of the result is important. Using (11) alters the sign of $c(F_{jp}, F_{ip})$ resulting in a positive value where c results in same direction as ϵ and a negative value where c is in the opposite direction.
$\psi(j,i,p,\epsilon) =$

$$\begin{cases} |c(F_{jp}, F_{ip})|, & \text{if } c(F_{jp}, F_{ip}) < 0 \text{ and } \epsilon = -1 \\ c(F_{jp}, F_{ip}), & \text{if } c(F_{jp}, F_{ip}) < 0 \text{ and } \epsilon = 1 \\ -c(F_{jp}, F_{ip}), & \text{if } c(F_{jp}, F_{ip}) > 0 \text{ and } \epsilon = -1 \\ c(F_{jp}, F_{ip}), & \text{if } c(F_{jp}, F_{ip}) > 0 \text{ and } \epsilon = 1 \end{cases} \tag{11}$$

This shall be referred to as the degree of *fitness*. That is, the degree to which i is a good recommendation on j for the attribute p. This results in a value within $[-1, 1]$.

3) For the final result, subtract the degree to which items do not fit (10) from the degree to which they do fit (11). The item with the largest resulting value is the best comparison. The ranking of item i against item j where attribute p is preferred to be in the direction ϵ (where $\epsilon \in \{-1, 1\}$) is therefore

$$f(j,i,p,\epsilon) = \psi(j,i,p,\epsilon) - \delta(j,i,p) \tag{12}$$

TABLE III
NOTATIONS USED THROUGHOUT THIS PAPER

Notation	Meaning
i	an item $i \in I$
q	an attribute $q \in Q$
t_Q	total number of attributes
F_{iq}	FS of attribute q for item i
F_{iQ}	the set of all FSs of all attributes $q \in Q$ for item i
j	the initial item that is compared against
p	the attribute the user wishes to change
ϵ	the direction of change invoked on p
W	an ordered set of weights
Notation	**Function**
s	similarity measure
d	distance measure
c	proximity measure (fusion of similarity and distance)
δ	measure of how much two items are similar
ψ	measure of how much two items differ according to p and ϵ
f	resulting ranked value by comparing j against another item

From (12), the highest valued result represents the best recommendation. The recommendations with high levels of fitness and low levels of unfitness result in the highest values, and therefore the highest ranks. Additionally, an item that has high fitness but also high unfitness will be ranked lower than an item that has high fitness and low unfitness.

The next section introduces extensions of the above method in which multiple attributes may be changed in measuring the degree of fitness, and attributes may be weighted.

B. Extensions of the Recommendation Method

1) Weighting the Attributes of Unfitness: The attributes for the degree of unfitness can be unevenly weighted so that one attribute may be given more importance than another. For example, one may wish to find a restaurant with a similar quality of both service and food, but placing more importance on food being similar rather than service. To achieve this, the level of unfitness is calculated as

$$\delta(j,i,p,W) = \sum_{q \in Q; q \neq p} W_q \; |c(F_{jq}, F_{iq})| \tag{13}$$

where W is the set of weights for each attribute, such that $\sum_{q \in Q} W_q = 1$, and W_p is set to 0. The weight W_p is excluded because it does not contribute to the degree of unfitness. Using (13), one may give preferences towards attributes which one wishes to keep similar. This is useful where some attributes are of greater importance than others.

2) Using Multiple Attributes of Fitness: A user may wish to change multiple attributes, for example, to find a restaurant with both a nicer ambience and a better quality of service. One simple way to accommodate for multiple changes is to average their results. The measure of fitness then becomes

$$\psi_m(j,i,P,E) = \frac{1}{N} \sum_{n=1}^{N} \psi(j,i,P_n,E_n) \tag{14}$$

where P is a list of the attributes to be changed, E is a list containing the corresponding direction (-1 or 1) for each attribute, and N is the total number of attributes to be changed.

When (14) is used, the measure of unfitness must be adjusted to exclude each attribute which is to be changed. The parameter p becomes the list of attributes P, and the summation changes to exclude all values in P. This, consequently, changes (10) to

$$\delta(j,i,P) = \frac{1}{t_Q - N} \sum_{q \in Q; q \notin P} |c(F_{jq}, F_{iq})| \qquad (15)$$

where N is the total number of attributes within P.

3) Weighting Multiple Attributes of Fitness: A user may also wish to weight multiple changes to find, for example, a restaurant with both a nicer ambience and a better quality of service, but placing more importance on the change in service than the change in ambience. To achieve this, the measure of fitness becomes

$$\psi_m(j,i,P,E,W) = \sum_{n=1}^{N} W_n \psi(j,i,P_n,E_n) \qquad (16)$$

where W is a list containing the corresponding weight for each attribute and $\sum_{i=1}^{N} W_i = 1$. Using multiple weighted attributes to calculate item fitness also requires (15) to calculate item unfitness. Note that when weighting unfitness (15) or fitness (16), it is important that the weights sum to 1 to ensure that the measures return a result within $[0,1]$ and $[-1,1]$, respectively.

The proposed method of calculating recommendations is dependent on the user's ability to describe their preferences in relation to an existing product. The next section demonstrates this approach within such contexts.

V. Demonstrations

First, a demonstration of the proposed method in Section IV-A is presented, followed by an example of why the combined measure is used in measuring both unfitness and fitness. Finally, demonstrations of weighting attributes, as shown in Section IV-B, are given. These demonstrations use synthetic data for simplicity, but future work will focus on real data.

A. Demonstration of the Proposed Method

An example using polygons of identical perimeter length and differing areas is used to demonstrate the recommendation method. By using this simple example with ground truth, it is easier to follow the process and judge what should be the expected results.

It is well known that given the same perimeter length, as the number of sides of a polygon increases the area within the polygon also increases. Table IV shows this for polygons of three to eight sides, each with a perimeter of ten where the area and perimeter both use the same scale. Examples for each polygon are shown in Fig. 4.

In this example, an approximation of the total number of sides and area of a polygon is used, in which the FSs could, for example, originate from human visual assessment; Fig. 5 shows a synthetic example of such FSs. Each FS has a Gaussian membership function with the mean at the number of sides and standard deviation of 1, and a mean at the total area with a standard deviation of 0.1.

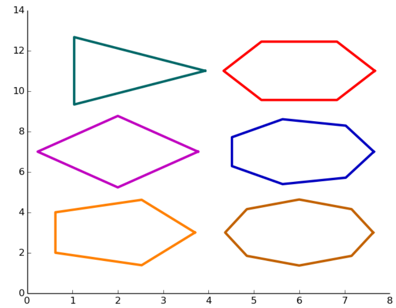


Fig. 4. Regular polygons from 3 to 8 sides with a perimeter length of 10.

TABLE IV
Area of regular polygons give a perimeter of length 10.

shape	area
triangle	4.8113
square	6.25
pentagon	6.8819
hexagon	7.2169
heptagon	7.4161
octagon	7.5444

The following demonstrates the results of using the method in Section IV-A to query:

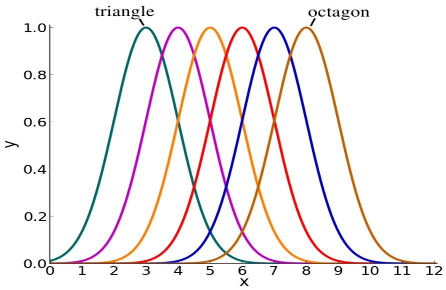
$$\begin{array}{l} \text{SELECT } polygon \text{ FROM } polygons \text{ AS } p \\ \text{WHERE } p.sides \text{ is similar to } hexagon.sides \\ \text{AND } p.area < hexagon.area \end{array} \qquad (17)$$

The results of the combined measure (9) in comparing a hexagon against other polygons for both attributes (total sides and area) are shown in Table V. Note that the first three polygons have fewer sides and a smaller area, and so their comparisons result in a negative value. In each comparison, the base comparison is always given as the first attribute of the directional distance measure; in this case, the base comparison is the hexagon.

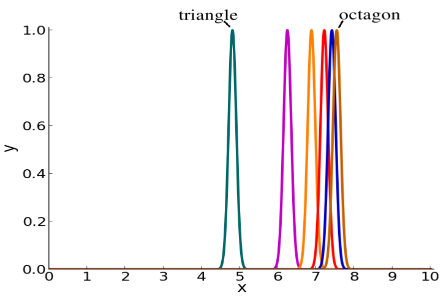
The results of steps 1, 2 and 3 of ranking recommendations are shown in Table VI. The degree of unfitness, which is obtained from step 1 (10) uses the absolute value of the comparison on the number of sides. In this case, only one attribute is used in the value of unfitness (the number of sides). If multiple attributes are used then their average value is used to represent unfitness. As the aim is to find the polygon with a smaller area than a hexagon, within the fitness score (11) the value ϵ is -1 because negative results of the combined

TABLE V
Comparison of number of sides and area of different polygons against a hexagon using (9).

shape	sides	area
triangle	-0.726	-0.772
square	-0.619	-0.729
pentagon	-0.409	-0.686
hexagon	0.0	0.0
heptagon	0.409	0.578
octagon	0.619	0.678



(a)



(b)

Fig. 5. (a) The number of sides of each shape represented as an approximation. (b) The area of each shape represented as an approximation. In each sub-figure, the left-most FS represents the attributes of a triangle, and the total number of total sides of the polygons increases towards the right, where the right-most FS represents the attributes of an octagon.

measure are favoured over positive results. Given this, the signs from the combined measure of similarity and distance (9) are reversed by (11) such that a negative value becomes positive (to represent a favourable recommendation) and a positive value becomes negative (to represent that the direction is unfavourable). The final result is created by subtracting the unfitness of an item from its fitness, and the resulting ranks are shown in Table VI.

From the values in Table VI, it is clear which results represent good recommendations. The positively ranked values are ranked higher than the hexagon (the base comparison) and represent 'good' recommendations. However, the heptagon and octagon give negative values indicating that they are 'poor' recommendations. Observing these results, a pentagon is the best recommendation. This is intuitively the expected recommendation because it has the most similar number of sides to a hexagon whilst also having a decreased area. Although the square and triangle have a greater decrease in area, their number of sides is more dissimilar and therefore they do not fit the query (17) as well as a pentagon. As a result, the square is the second best recommendation as it is the second most similarly sided shape and also has a decrease in area. The triangle is the least recommended because, although it has the greatest decrease in area and therefore the greatest fitness, its total number of sides is so different from a hexagon that its rank becomes lower than a square or pentagon. In summary, this synthetic example demonstrates the mechanism for calculating recommendations from complex queries and the results returned are those intuitively expected.

The next section demonstrates why dissimilarity and normalised distance cannot be used separately for unfitness and fitness, respectively, instead of using the combined measure.

TABLE VI
RANKING POLYGONS AGAINST A HEXAGON FOR LESS AREA.

shape	unfitness (10)	fitness (11)	result (12)	rank
triangle	0.726	0.772	**0.046**	**3**
square	0.619	0.729	**0.110**	**2**
pentagon	0.409	0.686	**0.277**	**1**
hexagon	0.0	0.0	**0.0**	**-**
heptagon	0.409	-0.578	**-0.986**	**4**
octagon	0.619	-0.678	**-1.30**	**5**

TABLE VII
RANKING POLYGONS AGAINST A HEXAGON FOR LESS AREA USING SEPARATE MEASURES.

shape	unfitness (4)	fitness (7)	result 12)	rank
triangle	0.93	0.2406	**-0.6894**	**3**
square	0.8131	0.0967	**-0.7164**	**4**
pentagon	0.5484	0.0335	**-0.5149**	**1**
hexagon	0.0	0.0	**0**	**-**
heptagon	0.5484	-0.0199	**-0.5683**	**2**
octagon	0.8131	-0.0327	**-0.8458**	**5**

B. Using the Combined Measure

It is important to use the same measure for both fitness (11) and unfitness (10) to ensure the results are meaningful (e.g. the proximity measure (9) is used in both calculations). To demonstrate this, a comparison is given using the method outlined in Section IV with different measures. In this example, the unfitness of two items is measured by their dissimilarity (4), and the fitness is given by using the normalised directional distance measure (7). Using these measures, the same example query as earlier is used (17). The results of these comparisons are shown in Table VII.

To compare using the combined measure (9) against using dissimilarity (4) and distance (7) separately, the final results are shown side-by-side for both methods in Table VIII. It is clear that both the values and the ordering are different for each method. When using the combined measure, positive results represent 'good' recommendations and negative results represent 'poor' recommendations. However, using separate measures results in a negative value for each comparison, indicating that every recommendation is a 'poor' one. These results are unclear because it is not meaningful to subtract two

TABLE VIII
SIDE BY SIDE COMPARISON OF RANKING POLYGONS AGAINST A HEXAGON FOR LESS AREA USING COMBINED AND SEPARATE MEASURES.

Combined measure		Separate measures	
shape	result	shape	result
0.277	pentagon	0	hexagon
0.110	square	-0.5149	pentagon
0.046	triangle	-0.5683	heptagon
0.0	hexagon	-0.6894	triangle
-0.986	heptagon	-0.7164	square
-1.297	octagon	-0.8458	octagon

fundamentally different measures from each other.

The rank order from using separate measures is also unexpected. The heptagon is ranked higher than a square or triangle despite having a larger area. This is not an ideal recommendation given that the results are intended to rank shapes with *less* area than a hexagon, as well as having a similar number of sides.

This demonstration shows that it is important to use the same measure in determining fitness and unfitness in comparisons to ensure meaningful results. The results from Section III also show that it is useful to use both similarity and distance, rather than choosing just one of the two measures, to gain an accurate comparison of FSs in determining recommendations. Therefore, the fused measure of similarity and distance (9) is an ideal measure in determining both the fitness and unfitness in calculating recommendations.

Next, demonstrations are given for the extensions in Section IV-B, in which attributes may be weighted.

C. Weighting Attributes for Unfitness

A user may feel that some attributes are more important than others and will therefore want to weight the importance of these attributes to affect the results of the recommendation process. To demonstrate the effects of using weighted attributes a small synthetic data set is used which describes, using FSs, the *food quality*, *affordability* and *service quality* of four restaurants on a scale from 1 to 10. The restaurants are labelled A, B, C and D. Each FS is represented by a Gaussian membership function, for which the mean value is listed in Table IX for each restaurant and attribute. For simplicity, the standard deviation of each FS is 1.

To demonstrate the effects of using weights, the same query is calculated twice, once with evenly distributed weights for the level of unfitness, and then with unevenly distributed weights. The goal is to find a restaurant with a higher *food quality* than restaurant A, but with similar *affordability* and *service quality*. Table X shows the results of this for each step from Section IV-A when applying equal weight to all attributes by using (10) to calculate the level of unfitness. In taking this approach, restaurant C is the best recommendation, followed by B. Although B has the highest quality of food, C is a better match against A because its other attributes are rated the same as A. Restaurant D, however, has a negative value and therefore is not a good recommendation against A. This is clear from the data in Table IX, which shows that D has a lower rating of *food quality* than A.

If the weights for the measure of unfitness are changed to place more importance on maintaining *service quality* over *affordability* then the results change. In this example, *service quality* is weighted at 0.8 and *affordability* is weighted at 0.2. To calculate the degree of unfitness for each restaurant, (13) is used to measure item unfitness instead of (10). The results of this process are shown in Table XI. Note that the degree of fitness has not changed from Table X because it is not affected by the weights.

TABLE IX
MEAN RATING OF FOUR RESTAURANTS A TO D FOR THREE DIFFERENT ATTRIBUTES.

restaurant	food quality	affordability	service quality
A	7	7	7
B	9	6	7
C	8	7	7
D	5	8	5

TABLE X
RESULTS OF RECOMMENDING A RESTAURANT THAT HAS A HIGHER QUALITY OF FOOD THAN A BASED ON THE DATA IN TABLE IX AND USING EVENLY DISTRIBUTED WEIGHTS TO MEASURE THE DEGREE OF UNFITNESS.

restaurant	unfitness (10)	fitness (11)	result (12)	rank
A	0.0	0.0	**0.0**	**-**
B	0.3013	0.7508	**0.4495**	**2**
C	0.0	0.6029	**0.6029**	**1**
D	0.6769	-0.7511	**-1.428**	**3**

In this example, B is now the most highly rated and C is the second highest. By giving a small weight to affordability, the fact that the affordability of B is different to A becomes less important. Additionally, due to the service quality of B being the same as A, and the food quality being better than C, the restaurant B becomes the best recommendation. This demonstrates that using weights is an effective method of assigning different preferences to different attributes.

D. Weighting the Attributes for Fitness

The following demonstrates using multiple changes in fitness attributes (16) using the data from the previous restaurant example. In this example, the aim is to find a restaurant which has both a higher *food quality* and a higher degree of *affordability* than restaurant A.

First, the *food quality* and *affordability* are equally rated using (14) to measure each restaurant's fitness, where P =[*food quality*, *affordability*], $E = [1, 1]$ and N = 2. Unfitness is calculated using (15). The results of this process are shown in Table XII, which indicate that C is certainly the best recommendation and B is also a possible recommendation. B's result is lower than C because, although it has an increase in *food quality*, it has a decrease in *affordability*.

Next, an example of weighting the fitness attributes is given. The weights $\{$*food quality*:0.3, *affordability*:0.7$\}$ are given, thus placing higher importance on an increase in restaurant

TABLE XI
RESULTS OF RECOMMENDING A RESTAURANT THAT HAS A HIGHER QUALITY OF FOOD THAN A BASED ON THE DATA IN TABLE IX AND USING UNEVENLY DISTRIBUTED WEIGHTS (*affordability*:0.2, *service quality*:0.8) TO MEASURE THE DEGREE OF UNFITNESS.

restaurant	unfitness (13)	fitness (11)	result (12)	rank
A	0.0	0.0	**0.0**	**-**
B	0.1205	0.7508	**0.6303**	**1**
C	0.0	0.6029	**0.6029**	**2**
D	0.7215	-0.7511	**-1.4726**	**3**

TABLE XII

RESULTS OF RECOMMENDING A RESTAURANT THAT HAS A HIGHER QUALITY OF FOOD AND AFFORDABILITY THAN A BASED ON THE DATA IN TABLE IX AND USING EVENLY DISTRIBUTED WEIGHTS TO MEASURE THE DEGREE OF FITNESS.

restaurant	unfitness (15)	fitness (14)	result (12)	rank
A	0.0	0.0	**0.0**	**-**
B	0.0	0.0741	**0.0741**	**2**
C	0.0	0.3015	**0.3015**	**1**
D	0.7511	-0.0741	**-0.8252**	**3**

TABLE XIII

RESULTS OF RECOMMENDING A RESTAURANT THAT HAS A HIGHER QUALITY OF FOOD AND AFFORDABILITY THAN A BASED ON THE DATA IN TABLE IX AND USING THE WEIGHTS (*food*:0.3, *affordability*:0.7) TO MEASURE THE DEGREE OF FITNESS.

restaurant	unfitness (15)	fitness (16)	result (12)	rank
A	0.0	0.0	**0.0**	**-**
B	0.0	-0.1966	**-0.1966**	**2**
C	0.0	0.1809	**0.1809**	**1**
D	0.7511	0.1967	**-0.5544**	**3**

affordability. In this case, (16) is used to calculate each restaurant's fitness, where P and E are the same as earlier, and $W = [0.3, 0.7]$. The results of this are shown in Table XIII. In this example, B has become a poor recommendation because it has a decreased rating of *affordability*. Restaurant C also has a lower result and thus less confidence as a recommendation. This is because there is no change in *affordability* between restaurants A and C, so the fitness score only includes the weight 0.3 for the increase in *food quality*. In the previous example, however, the weight 0.5 was used for the increase in *food quality*, making C a more confident recommendation.

Examples have been given to show the effects of weighting attributes in determining the unfitness and fitness of items. Although only weighting one comparison (of unfitness and fitness) at a time has been demonstrated, it is possible to weight both comparisons to express preferences in both unfitness and fitness to find recommendations.

VI. CONCLUSIONS

A method of enabling complex queries on fuzzy data is introduced, in which users are given recommendations on items (or products) based on imprecise queries such as "I want a restaurant similar to this one but with a nicer ambience." Such queries allow users to search for an item when the user has only a vague idea of his/her preferences. Based on this type of query, recommendations are made by breaking down attributes into two classes, those that should be similar to a given item and those that should be different. The comparisons of these attributes are performed through a combination of similarity and distance measures. Experiments show that the method is effective at finding and ranking recommendations based on such imprecise queries. Methods of weighting attributes are also introduced, allowing a user to indicate preferences on specific attributes over others; this is demonstrated to effectively alter the results.

The proposed method allows for expansions into more complex methods of recommendations. For example, it can easily be used to build recommendations from a user profile. The base item, which is compared against, may be a collection of FSs representing the user's preferences, or a combination of this and the information of a known product. Collaborative recommendations may also be introduced where user profiles are known. In this case, the history of other users may influence the fitness or unfitness of items.

Future work will compare the proposed method to other related methods within the literature. Additionally, focus will be given to applications of this approach to larger data sets containing real data which has been collected by surveys using the Interval Agreement Approach [1].

REFERENCES

[1] C. Wagner, S. Miller, J. Garibaldi, D. Anderson, and T. Havens, "From Interval-Valued Data to General Type-2 Fuzzy Sets," *IEEE Trans. Fuzzy Syst.*, vol. PP, no. 99, pp. 1–1, 2014.

[2] L. Zadeh, "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 2, pp. 103 –111, May 1996.

[3] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete, "A collaborative recommender system based on probabilistic inference from fuzzy observations," *Fuzzy Sets and Systems*, vol. 159, no. 12, pp. 1554–1576, 2008, advances in Intelligent Databases and Information Systems.

[4] A. Zenebe and A. F. Norcio, "Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems," *Fuzzy Sets and Systems*, vol. 160, no. 1, pp. 76–94, 2009, theme: Aggregation Operations.

[5] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 61–82, 2002.

[6] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston *et al.*, "The YouTube video recommendation system," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 293–296.

[7] M.-H. Hsu, "A personalized English learning recommender system for ESL students," *Expert Systems with Applications*, vol. 34, no. 1, pp. 683–688, 2008.

[8] Y.-F. Wang, Y.-L. Chuang, M.-H. Hsu, and H.-C. Keh, "A personalized recommender system for the cosmetic business," *Expert Systems with Applications*, vol. 26, no. 3, pp. 427–434, 2004.

[9] Y. Cao and Y. Li, "An intelligent fuzzy-based recommendation system for consumer electronic products," *Expert Systems with Applications*, vol. 33, no. 1, pp. 230–240, 2007.

[10] J. McCulloch, C. Wagner, and U. Aickelin, "Analysing fuzzy sets through combining measures of similarity and distance," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, July 2014, pp. 155–162.

[11] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[12] A. Lipkus, "A proof of the triangle inequality for the Tanimoto distance," *Journal of Mathematical Chemistry*, vol. 26, no. 1-3, pp. 263–265, 1999.

[13] B. Chaudhur and A. Rosenfeld, "On a metric distance between fuzzy sets," *Pattern Recognition Letters*, vol. 17, no. 11, pp. 1157–1160, 1996.

[14] J.-S. Yao and K. Wu, "Ranking fuzzy numbers based on decomposition principle and signed distance," *Fuzzy Sets and Systems*, vol. 116, no. 2, pp. 275–288, 2000.

[15] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 1, pp. 183–190, 1988.